具有隐私保护的访问控制系统的设计与实现

# **1、绪论**

本章主要介绍课题开发的研究背景及意义，以及目前国内外的研究现状；并对主要工作内容和文章整体结构进行概述。

## 1.1.课题开发研究背景

随着网络化的程度不断地提高，在网络上获取信息也越来越方便容易。近些年来，随着大数据的兴起和发展，数据成为了许多大型企业重要资产的一部分，大数据能够为企业进一步理解客户需求，改善技术并且促进产品升级提供重要的参考依据。通过不断进行大数据的研究，将技术与大数据深度融合，这也成为了许多企业升级的新动力，许多传统企业已经实现了转型升级，同时也有大量企业在为之努力。总之，在当今时代，数据中蕴含着很多的生产力和财富，可以说，谁能够先一步掌握数据，就能够掌握进步的先机和主动权。

近些年来，越来越多的企业都开始采用大数据的分析技术。例如，商业，医疗卫生业，以及许多科研项目。利用大数据分析可以做出更好的决策。比如，现在网上流行的各种购物网站（如京东，天猫），为了应对在6.18，双11等购物节中客户大量购物所带来的供货不足、跨区调货慢等风险，这些企业都开始采用大数据分析技术，通过观察用户经常浏览、加入购物车或者收藏的商品来进行提前的调货备货，有效的解决了购物量暴增的问题。再比如，法国Orange旗下的电信公司Telekomunikacja Polska是波兰最大的语音和宽带网络供应商，他们通过对客户进行细分，观察大量的客户的数据记录，重点关注“谁经常给谁打电话”以及“打电话的时长和频率”这两种信息。从而勾勒出一张社交网，通过这个所谓的“社交图谱”可以较为准确的预测哪种客户的流失将会导致更大的损失，从而优先采取措施挽留这一类的用户。值得注意的是，运用这个方法分析出的客户流失模型准确率高达47%，成功的帮助该电信公司挽留多数核心用户。

然而，任何事情都具有两面性。数据分析是一把双刃剑，一方面我们可以通过精准的数据分析为我们带来可观的收益，在另一方面，这也存在着许多的潜在风险。要想分析出准确的模型，做出准确的决策，必然要利用大量的数据，势必要通过许多有偿或者无偿的数据共享来实现。在数据共享访问的过程中，如果对访问授权的管理出现问题，大数据所带来的威胁将会是前所未有的。近几年来，随着技术的飞速发展，隐私保护的访问控制系统也随之发生了重大的改变。

## 1.2 课题开发研究意义

随着近些年来互联网的普及程度不断地提高，人们在享受着共享信息带来的无限便利的同时，也面临着敏感信息和个人隐私等重要的资料泄露的问题。特别是近些年来，随着技术的进步，我们的个人信息正在被各类主体挖掘和分析，我们在生活中或多或少都会遇到这样的现象：在刚查询过贷款信息后就会有很多的所谓金融公司打来骚扰电话；注册完网上教育平台账号后就会有许多教育机构推荐广告发到你的邮箱。这样的例子比比皆是，或许这些现象看起来并不起眼，但是，因隐私信息流失而带来的侵权、诈骗等犯罪行为越来越多，这已给我们的社会带来了很大的损失，也严重影响着社会的安宁。

根据调查分析发现，泄露信息的各种类型中，绝大多数以个人隐私和商业的机密为主，占所有的泄露信息95%以上。而且，泄露信息仍然呈上升的趋势，更为严峻的是，当今泄露信息的手段已经不止于以前的黑客入侵等手段，近几年，非技术手段的运用迅速的增长，这也需要引起我们的警惕。敏感信息的泄露所涉及的行业范围也越来越广泛，但目前仍然主要集中在互联网，金融，制造业行业和政府机构。一篇来自《中国青年报》的研究显示，被调查的随机样本中，超过55%的人认为在如今高度发达的互联网时代，很难保证自己的隐私不被窃取，很难保证个人隐私的安全；12.3%的人在互联网上注册个人信息时，不愿将个人真实信息填报上去；15.1%的人表明在工作场所的行为被监视。《纽约时报》中发表的文章称，如今互联网上隐私保护的技术已经无法赶上其他技术的变化和客户的需求。如何在网络环境中，有效的保护用户的隐私信息，已经逐渐成为新时代网络服务质量和用户满意程度的重要判定标准。

访问控制系统着眼于设计并实现一个安全的、合理的信息管理系统。通过设置一些控制的机制和策略，控制主体的请求和行为等操作。将所有恶意用户的访问要求拦截、并且监督所有用户可能发生的恶意行为。访问控制系统对为不同的用户、不同程度的敏感信息设置不同的机制和策略至关重要，也是提高用户对当今网络的信任度和接受程度的重要一环。

## 1.3 国内外研究现状

早在20世纪70年代，便产生了访问控制技术。在当时，一些应用于商业的计算机中逐渐出现了访问控制技术的雏形。一开始主要是为了研究解决在大型计算机系统中共享数据的授权管理问题。访问控制技术一经提出，便得到了业内的广泛响应，并通过40多年来的蓬勃发展，随着访问控制技术的不断完善，一批批具有代表性的技术逐渐出现在世人面前，访问控制技术的应用范围也随之迅速扩大。对访问控制模型的研究也取得了相当可观的成果。

20世纪70年代，随着初期的分时操作系统的问世，Jampson等人为了解决当时多用户计算机系统的数据保护问题，提出了最早的访问控制模型即自主访问控制模型（DAC），其基本思想是将系统划分为两种不同类型的角色：主体、客体。通过建立一张它们之间的关联表，由主体自主的授权，控制不同主体的访问权限。

与DAC同时期的另一种早期的访问控制模型是强制访问控制模型（MAC），这一模型最早是由美国的政府以及军方所提出的，最初的目的是出于抵挡特洛伊木马等病毒的入侵以及对不同军事信息机密性的不同需求。其基本思想是为不同的客体设置不同的敏感程度以及安全属性，通过比较主体和客体之间不同的安全属性级别，来判断该主体是否有权限访问客体。在大型多层次安全级别的系统中常会应用到此访问控制技术。

在20世纪90年代，Ferraiolo 和 Kuhn 提出了基于角色的访问控制模型（RBAC），实际上早在20世纪70年代，一些基于角色的访问控制技术便被应用到一些商用计算机系统之中。于1992年，正式提出了基于角色的访问控制概念。该模型引入了角色的概念，一个用户可对应一个角色也可分配多个不同的角色。为各个角色设置不同权限后实现访问控制。这一模型简化了权限管理，将用户和访问权限进行逻辑分离，降低了它们之间的耦合程度。

目前仍有多个访问控制模型尚处于研究发展阶段。随着近年来网络分布式计算的快速发展，使得计算机任务的组织推进进一步自动化，这也促使人们将对系统安全方面问题的考虑从以往的计算机中静态的主体和客体逐渐向动态的随着任务的推进，动态的改变权限。

1997年，Tomas提出了一种基于任务的访问控制模型（TBAC），他所采用的是一种“面向任务”的观点，从计算机所要执行的任务的视角出发建立安全机制和实现这种机制的安全模型，在任务的推进过程中，对系统进行动态、实时的权限等方面的管理。

在公共开放的环境（如互联网）中，服务器端需要不断地响应客户端的频繁交互请求，这些请求交互往往处于不同的安全域之内，传统的基于角色、身份的访问控制模式已经不能满足于这种环境，这时出现了基于属性的访问控制系统（ABAC），这种模式能够很好的适应开放的网络环境。从基于属性的访问控制模型角度出发，访问权限的判定基于请求客体和资源所具有的属性，请求者和资源的属性通过一些特性来标识。这个模型具有很好的可扩展性以及灵活性，他提供了网络环境下安全匿名访问的实现可能，这在当今大型的分布式环境下是很重要的一点。

综上，在过去的五十年的发展历史中，访问控制模型已经取得了许多丰硕的成果。每种模型都有其自身特点以及适用范围，这部分我们将在第二章的内容中详细分析。

## 1.4 主要工作内容及论文组织结构

本章主要介绍课题开发的研究背景及意义，以及目前国内外最新的研究现状；最后，将介绍作者所进行的主要工作内容以及论文的组织结构。研究的现状进行分析。随后在介绍了一些与隐私保护相关的概念的基础上，对现有的一部分访问控制模型进行介绍并且将他们的优缺点进行对比分析。最后，将详细介绍一个作者设计的具有隐私保护的访问控制系统，这个系统主要是为了实现不同个性化需求的隐私保护，将主体的隐私信息作为重要因素加以考虑并且在传统模型的基础上加了一些自己的模块。

论文组织结构

1. 主要介绍课题开发的研究背景及意义，以及目前国内外的研究现状；最后，对作者所进行的主要工作内容进行描述以及对全篇论文的组织结构进行阐述。
2. 对文中可能涉及的名词术语进行说明介绍（尹绍锋，2008）。随后，本章介绍了一些目前比较流行的访问控制模型，并且对不同模型的优劣进行对比分析。
3. 在之前分析了几种常见的访问控制模型及优缺点的基础之上，设计并实现一个个性化的具有隐私保护的访问控制系统，此系统主要考虑到隐私主体的隐私信息，并将隐私信息作为访问控制的重要因素加以考虑。除此之外，在这些传统模块的基础上，还增加了对主体行为的控制模块。

本章用UML建模语言在对访问控制系统进行概述的基础上，介绍了系统的需求分析，其中包括功能性需求分析、非功能性需求分析以及对系统的总体流程进行分析。

1. 本章在确认了访问控制系统的需求之后，开始介绍系统相关的设计问题。主要涉及对系统的体系结构选取、系统技术分析及选型、功能结构设计、数据库中的表设计及多表之间的关系设计。

# 2、相关知识介绍

本章主要对文中可能涉及的名词术语进行说明介绍（尹绍锋，2008）。其次，本章介绍了一些目前比较流行的访问控制模型，并且对不同的模型特点进行分析比较。

## 2.1 相关概念介绍

早在20世纪70年代，访问控制技术便由B.W.Lampson提出，初衷主要是为了研究解决大型计算机系统中那些共享数据的授权管理问题。该技术主要由三个元素构成：主体、客体、控制策略。三者之间的关系如图2-1所示。

1. 主体（subject）

主体即指对某种资源提出访问要求或请求动作的发起者，主体可以是用户行为的实体，常为用户或任何其他代理用户（相关利益人）发出的行为实体都可作为主体。

1. 客体（object）

客体即指被动实体，允许其他实体访问、操作。任何可以被读取、访问、修改等操作的资源、信息都可以作为客体。为请求者提供可使用的信息、服务和系统组件。常见客体有：资源、信息、文件、数据库。

1. 操作（operation）

泛指主体对客体的一种行为。是在主体、客体之间信息流动的方式。如果是对文件的操作，则可以分为，读、写、重定位、截断等。

1. 访问控制策略（Access Control Policy）

这是一个和授权相关的规则集，在这个规则集中将主体和客体之间可以被允许的作用行为和约束条件进行了定义。相关的权限不可以超出规则集合的范围

1. 许可（permission）

许可即指可以对某个客体进行的某种操作。

许可是由操作和客体组成的二元组，可以用描述操作与客体见的关系。其中代表许可；代表操作，如增加、删除、修改、查询等；代表客体，如文件、报表、数据库等。许可与权限（rights）含义相同，本文在不同场景使用不同术语。

1. 授权（authorization）

授权是指向主体授予某些特定的权限的过程。授权后，在已授权范围内，主体即可对客体进行访问操作。授权通常由系统安全人员发出，如为角色或用户分配权限等。

除了以上所述的几个概念，还有一些其他的相关概念，例如：认证、访问控制表、访问控制矩阵等，由于和本文研究内容关系不大，所以不在此处多加赘述。

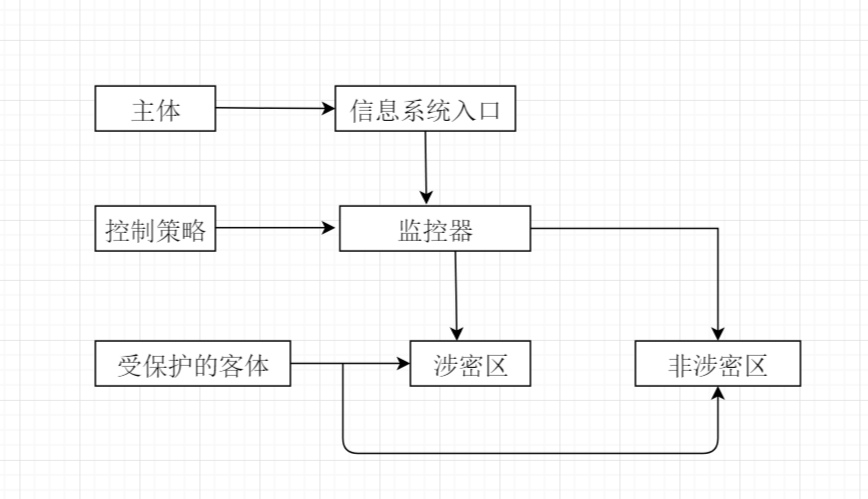


图2-1 主客体以及控制策略三者之间的关系

## 2.2 访问控制模型特点分析

### 2.2.1 自主访问控制模型（DAC）

20世纪70年代，随着初期的分时操作系统的问世，Jampson提出了自主访问控制模型，这一模型的提出主要是为了解决当时多用户计算机系统的数据保护问题。其基本思想是将系统分为两种不同的角色：主体、客体。通过建立一张它们之间的关联表，由主体自主的授权，控制不同主体的访问权限。DAC最大的优势就是他很灵活、容易实现、看起来较为直观。在访问控制矩阵（ACM）等技术支持的基础上使得每一个主体都能自主的将其自身所拥有的权限分配给其它任意主体，也可以将其回收，这一模型极大的分担了系统管理员的工作量，每一主体可根据需要在工作中设置不同的权限，极大的提高了权限设置的灵活性。但是这样的优势也会带来一些隐患：权限分配不需要管理员干预而采用自由分配的机制。系统安全管理员很快便会失去对系统权限分配的控制。在大型系统中，包括的主体以及客体的数量很大，使用这个模型会使得代价很大。同时，在这种访问控制模型下会有权限传递的现象出现，这也会带来隐患。

### 2.2.2 强制访问控制模型（MAC）

MAC是由美国的政府以及军方最早提出的，也是传统的访问控制系统之一。其主要目的源于对军事信息的机密性的不同要求以及为了组织特洛伊木马等病毒的入侵。其基本思想是为不同的客体设置不同的敏感程度以及安全属性，通过设置主体和客体之间不同的安全属性级别，为不同用户设置不同的可信任度级别，以此来判断该主体是否有权限访问客体。在MAC模型中，安全级别的设置具有强制性，只有系统的安全管理员可以对其进行权限分配设置，主体和客体的安全级别无法由用户自主的改变。MAC模型具有层次性，信息流在系统中只能按规定由较低安全级别的层次流向高安全级别层次，不可向上读也不可向下写。在大型多层次安全级别的系统中常会应用到此访问控制技术。MAC的安全属性由多个级别，常用的安全级别从上到下可定义为：绝密级，秘密级，限制级和无级别级。系统安全管理员为所有主体及客体分配不同的安全标签，用于标识其安全等级。

如图2-2所示为属于强制访问控制模型的两种常见模型 BLP和Biba，其中，BLP针对于保护数据的机密性，通过上读下写来保证敏感的数据不外泄。而Biba则是针对于数据的完整性，它通过下写上读，保证了重要的数据不被篡改。

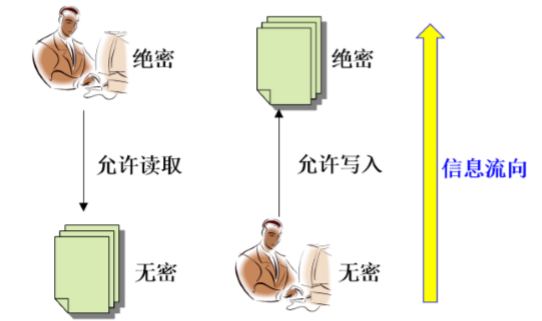
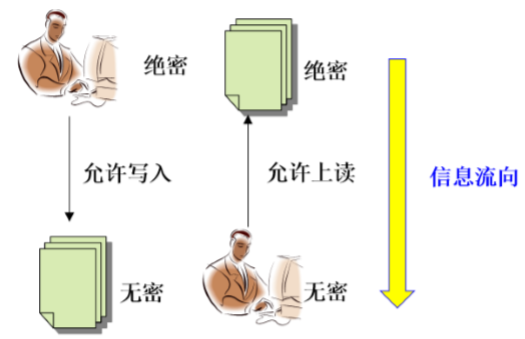
 

图2-2 BLP模型、Biba模型

MAC的优点在于它可靠性强，安全性较高，可控的程度较高，能够有效的防止特洛伊木马的攻击，防止隐私泄露；其缺点在于不够灵活，过于强调机密性；层次较多，实现的难度及工作量很大，难以管理，权限难以动态的变化。

### 2.2.3 基于角色的访问控制（RBAC）

1992年，Ferraiolo 和 Kuhn 提出了基于角色的访问控制模型的概念，该模型引入角色这一概念设置在用户和访问权限的中间，一个用户可以关联一个或多个角色，角色可以随着系统的执行动态进行创建和取消，并通过不同角色来实现对用户访问权限的管理。这一设计简化了权限管理，使得权限管理更加清晰，降低了用户和访问权限之间的耦合程度。如图2-3展示了一种用户、角色、访问控制的结构图。

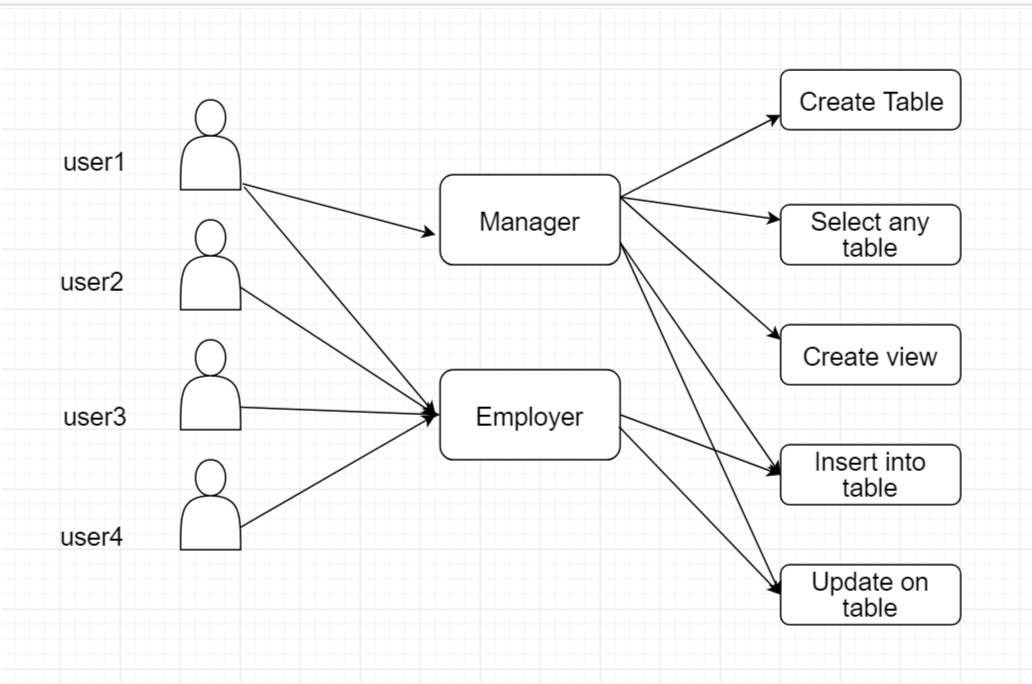


图2-3 一个简单角色的结构图

RBAC的优点在于通过新加了一层角色的概念，从而在逻辑上将用户和权限进行分离，可以描述静态、动态约束，容易管理，在很多系统中具有很强的实用性。缺点是权限的管理不够灵活，只对不同的主体进行了划分，不支持权限的动态管理，缺乏对客体的描述，可扩展性不强。从一个主体引发出多个角色，约束颗粒大，可能会造成一些安全隐患。没有提及环境的因素，不能精确的描述访问控制的过程。

### 2.2.4 基于任务的访问控制模型（TBAC）

1997年，Tomas提出了一种基于任务的访问控制模型（TBAC），这是一种新兴的访问控制模型。他站在应用或企业等角度来解决一些安全问题。与传统的系统角度不同，他所采用的是一种“面向任务”的观点，从计算机所要执行的任务的视角出发，建立安全机制和实现这种机制的模型，在任务的推进过程中，动态、实时的对系统进行授权等方面的管理。随着任务执行过程中的环境变化，对象的访问权限随之发生相应的动态变化，所以他也被称为主动安全模型。如图2-4展示了一个TBAC模型结构图。

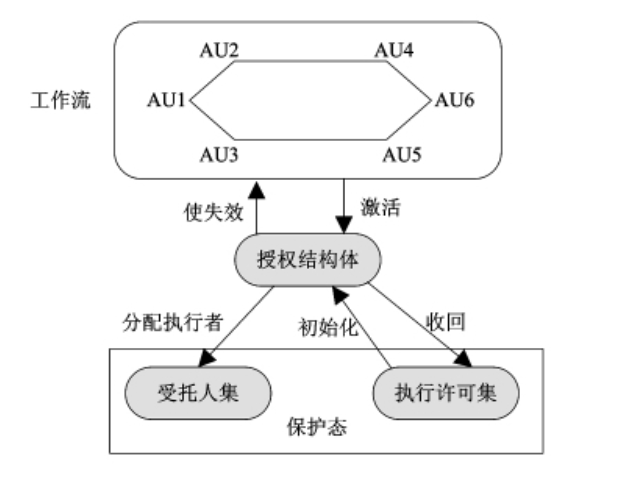


图2-4 TBAC模型

TBAC的优点在于他引入了任务的概念，可以实现授权的动态管理，权限随着任务的变化而变化，适用于分布式计算机系统。其缺点在于对主体的管理不够灵活，没有角色的概念，无法将角色和任务进行明确分离，缺乏对被动访问操作的控制。

### 2.2.5 基于属性的访问控制（ABAC）

基于用户（请求者）、被访问资源、访问操作和上下文属性提出的基于属性的访问控制模型将主体和客体的一些属性作为访问控制的决策要素。具有很大的灵活性，可以凭借主体具有的所有属性集合作为判断标准，决定该主体是否拥有访问权限，这种方法能够有效分离出权限判断和判定策略。同时，ABAC的管理相对简单，主体以及客体所具有的属性是其固有的，因此不需要管理员进行手动分配。这种模型本质上是一个多对多的模型，可以利用多个属性对一个实体进行描述，可以根据实际情况适量的做出调整。除此之外，ABAC模型具有很好的可扩展性，它支持同其他的隐私数据保护的机制相结合，保证用户的数据不被泄露。ABAC提供了网络环境下安全匿名访问的实现可能，这在当今大型的分布式环境下是很重要的一点。

ABAC模型的框架如图2-5所示，在ABAC的框架模型中，PEP接受外来的原始访问请求，并将请求发送到系统的属性权威（AA），在AA中生成基于属性的访问请求，并将请求发送到策略判定点，最终为PEP响应一个判定的结果，即能否访问相应的资源。

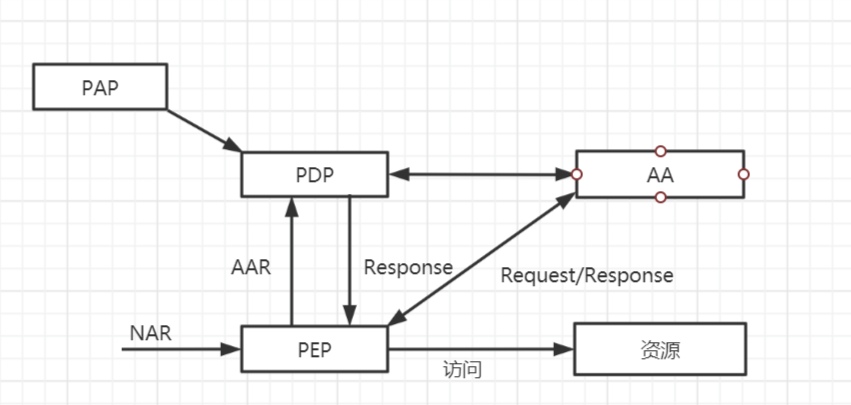


图2-5 ABAC模型框架示意图

ABAC的优点在于它是一个框架型模型，可以用于和其他的访问控制模型相结合，如RBAC。ABAC模型以属性为粒度，比角色更具有表达力。更加能够客观地表示访问控制过程。同时，属性具有通用性，在运行过程中，ABAC不必像RBAC那样创建一张用户-角色网络，创建大量的新角色并映射到相应权限，解决了RBAC中“角色爆炸”的问题。

ABAC的缺点在于它需要将主体以及客体的属性完整表达出来，但是将一些要素以属性的形式表达出来并不容易，策略的指定也随即变得更加复杂，语义问题也会使策略的制定出现冗余和冲突。

## 2.3 访问控制模型对比分析

上文对目前流行的各类常见访问控制模型进行了概括描述。在这一节，我们按照一些常规的考察指标，对以上所介绍的几种模型进行分析对比。为了看起来比较直观，以表格的形式呈现，如表2-1所示。表格中以“√”符号表示对应模型在哪些性能方面表现较好，若为空则表示此模型没有该特征或者在该方面表现不好。

表2-1 各类访问控制模型的分析比较

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 安全性 | 灵活性 | 权责分离 | 描述能力 | 上下文环境 | 动态变化/扩展性 | 可应用性 |
| DAC |  | √ |  |  |  |  |  |
| MAC | √ |  |  |  |  |  |  |
| RBAC |  | √ | √ | √ |  |  | √ |
| TBAC |  |  | √ |  | √ | √ | √ |
| ABAC |  |  |  | √ | √ | √ | √ |

## 2.4 本章小结

本章对访问控制技术的相关名词进行概念介绍。随后，挑选了几个典型的访问控制系统模型逐一介绍分析，并放在一起进行对比分析。可以看出，经过无数前辈近五十年的研究，对于访问控制模型的研究已经取得了相当可观的成果，访问控制技术有了很大的发展突破。同时，也可以看出，目前存在的各类访问控制模型都有着自身的特点及其使用范围。每种访问控制模型都有一些缺点和特有的使用环境。近年来随着计算机的普及程度不断提高，以及分布式系统等新系统层出不穷，这也给访问控制的研究带来了很多新的挑战。未来研究的方向将主要集中于优化、完善以及应用传统的访问控制模型，并且对新型访问控制模型进行探索。新的访问控制研究的领域或集中在可信计算、P2P、分布式网络等领域中。

# 3、访问控制系统需求分析

本章在之前分析了几种常见的访问控制模型及优缺点的基础之上，本文之后的部分将致力于一个个性化的具有隐私保护的访问控制系统的设计与实现，此系统主要考虑到隐私主体的隐私信息，并将隐私信息作为访问控制的重要因素加以考虑。除此之外，在这些传统模块的基础上，还增加了对主体行为的控制模块。

在对访问控制系统进行概述的基础上，本章用一些图形化建模语言对系统进行需求分析，其中包括功能性需求分析、非功能性需求分析以及对系统的总体流程进行分析。

## 3.1 系统总体概述

## 3.2 系统功能性需求分析

## 3.3 系统非功能性需求分析

## 3.4 系统总体流程描述

# 4、访问控制系统设计

本章在确认了访问控制系统的需求之后，开始介绍系统相关的设计问题。主要涉及对系统的体系结构选取、系统技术分析及选型、功能结构设计、数据库中的表设计及多表之间的关系设计。

## 4.1 系统体系结构分析

当前大部分web应用系统都采用C/S模式或者B/S模式（也有少数二者混合模式存在），C/S模式是一个比较古老而且技术十分成熟的架构，而B/S模式是从C/S模式中演变出来的一种新兴的架构。本节将详细介绍C/S、B/S的相关知识和各自的优缺点，在对两者进行比较之后，最终选择了B/S模式作为这个控制系统的架构。

### 4.1.1 C/S、B/S软件结构介绍

1、C/S结构

C/S结构即客户机（Client）/服务器（Server）模式。服务器负责对数据进行存储和处理，客户机主要负责处理和用户之间的交互工作。大量客户机可通过局域网与服务器获取连接，在获取到用户请求后，客户机连接服务器获取相关的数据，数据在传输到客户机之后由客户机进行整理，并将结果展示给最终用户。

2、B/S结构

B/S结构即浏览器（Browser）/服务器（Server）模式。是一种新兴的基于web的结构。客户端不再需要处理和用户之间大量的交互动作，只需要下载安装浏览器即可。客户端通过浏览器发送请求到服务器端，系统的主要功能的实现以及数据的存储都在服务器端，通过服务器端发送响应到客户端。该结构的优势在于客户端无需安装，简化了系统的开发、系统方便维护以及升级，也极大的减少了客户端的负载量，有效降低成本。

### 4.1.2 C/S、B/S软件结构的区别

B/S结构是从C/S结构改进而成的，B/S结构为三层架构，而C/S结构则为双层架构。由此可以看出他们之间存在密切的关系。近年来，伴随网络技术的迅猛发展，B/S结构从C/S结构中逐渐分离出来，成为当今主流结构之一。然而我们并不能片面的断定某一种结构更加优秀，也没有一种适用于所有开发设计的结构模式。B/S结构与C/S结构模式各具特色、各有优势。接下来以表格的形式对这两种结构进行比较：

表4-1 C/S结构与B/S结构比较

|  |  |  |
| --- | --- | --- |
|  | C/S结构 | B/S结构 |
| 硬件环境 | 建立在局域网的基础之上 | 在广域网的基础上建立，不需要专门的硬件支持 |
| 安全性能 | 采取点对点连接模式，数据处理基于不同的网络协议，安全性有保障 | 采取一对多、多对多的结构模式，具有很好的开放性，服务器需要做好数据库的安全管理，需要今后不断完善网络安全技术 |
| 可重用性 | 可重用性不如B/S好 | B/S结构具有多层结构，具有良好的重用性 |
| 维护难度 | 系统升级困难 | 升级开销较小，方便 |
| 请求处理 | 集中到服务器处理 | 分散处理 |
| 客户端接口 | 与操作系统有关 | 与浏览器相关 |
| 信息交互 | 交互性差 | 交互性强 |

总之，当具有复杂的业务流程，需要更高的系统响应速度以及要求用户操作界面有更个性化需求的系统时，C/S模式更加合适。而B/S模式则适用于客户分布广泛，需要更高灵活性的查询浏览功能，但业务处理逻辑相对简单的系统。

基于以上对B/S、C/S结构的分析比较，并结合第三章对访问控制系统的需求分析，本系统将采用B/S结构对系统进行实现。系统结构图如图4-1所示

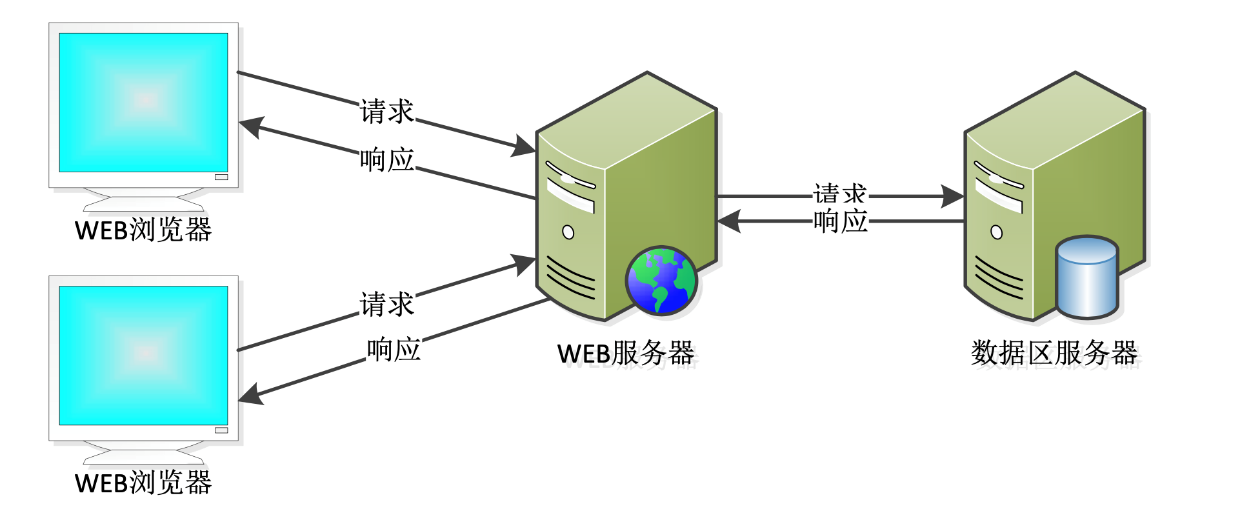


图4-1 系统结构图（B/S结构）

## 4.2 系统技术分析

### 4.2.1 MVC软件构件模式

针对访问控制系统日常工作中所面向的主要服务对象的特点进行分析，系统中主要涉及客户端浏览器、服务器、数据库三层体系结构。三者之间通过广域网进行通信，整个系统主要由与用户进行交互、业务逻辑操作和数据库对应操作等部分组成，故选取MVC设计模式来设计并实现此系统最为合适。MVC模式如图4-2所示。

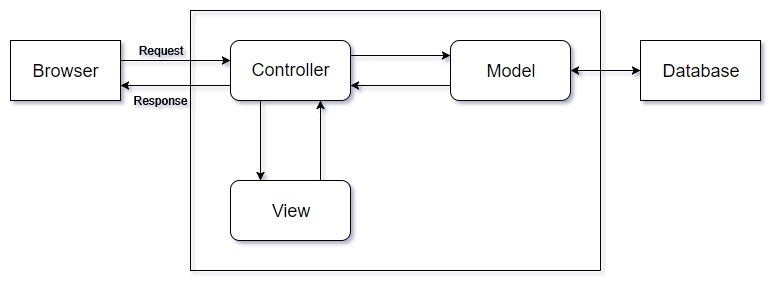


图4-2 MVC模式介绍图

1. View（视图层）

视图层是系统用于和用户进行交互，将对应的内容呈现给用户的部分。MVC设计模式将视图和业务逻辑进行分离，减小了耦合性，并且在处理多个视图和业务逻辑时代码的层次清晰明了。视图层只着眼于将从控制层得到的数据展示给用户，至于数据的正确性及数据从何而来，视图层不用考虑。这简化了开发人员的工作量，在开发的时候不需要过多担心视图层的改变会影响到其他层次。

1. Model（模型层）

在本系统中，模型层主要负责管理数据库和数据库的一些基础操作。该层次主要是实现对数据库的一些增删改查等基本操作，为不同的数据库设计接口以统一的方式设计，提高系统执行效率的同时能够使得系统处理大量数据的更新和查找。将视图和数据存储进行逻辑分离，不仅可以减少数据操作等方面代码的冗余，数据库的安全性也得到了保障。

1. Controller（控制层）

控制层是MVC架构中的主体。控制层主要负责将模型层和视图层进行连接，在控制层可以向模型层请求信息，调用模型层提供的服务，并将获得的反馈信息和数据发送到视图层，最终展示给用户。

MVC处理过程：客户端浏览器向服务器端发送请求，控制器接受用户发来的请求，调用相应的模型层所提供的服务，由模型层处理具体的业务逻辑，将处理后的数据返回给控制层，控制层再将反馈信息发送给视图层，将数据展示给用户，发送响应信息到浏览器。

MVC模式实现了系统的前后端分离，将业务逻辑分散到不同的层次中，层次结构清晰，每一模块都各司其职，具有很好的内聚性。这样的应用系统具有良好的稳定性，错误传播的范围较小，各个构件也可以在其他系统中复用。

### 4.2.2 Bootstrap、Spring Boot框架

框架是一种经过验证、具有一定功能的半成品软件，对一些基础的代码进行了封装，软件开发人员在使用框架时可以直接调用封装好的API，这样可以减少许多代码的编写，提高了软件开发的效率，简短了开发的周期。

1、Bootstrap框架

Bootstrap是一个用于前端开发的框架，由Twitter在2011年推出的一个开源WEB的前端框架，发布时间不长但是技术已经非常成熟，深受开发人员喜爱。Bootstrap 是基于 HTML、CSS、JavaScript 的，通过查询不同组件可以构建出非常优雅的前端界面，并且只占用很少的资源，使得 Web 开发更加便捷。

目前版本的Bootstrap框架主要包括完整的CSS样式和JavaScript插件，可以使用这些样式和插件得到丰富的页面效果。同时，Bootstrap框架采用响应式布局，他的实现依赖于栅格系统。使同一个页面可以在不同分辨率的设备中展示。

2、Spring Boot框架

在介绍Spring Boot框架概念之前，先对Spring框架的优点及缺点进行分析:

优点分析：Spring是Java企业版（J2EE）的轻量级产品，它为Java企业级开发提供了一套相对容易的办法，主要依赖于控制反转（IoC）和面向切面编程（AOP），利用简单的Java对象实现了EJB所需功能。

缺点分析：虽然该框架是轻量级产品，虽然实用，但是它所要求的回报也不少，在使用Spring进行开发的过程中，需要大量的时间用于配置，在考虑Spring配置和解决系统业务逻辑问题之间开发人员需要进行思维的切换，因此编写配置会占用许多编写系统的时间。此外，系统的项目依赖管理也是一个十分繁琐的步骤。在最初环境搭建时，需要导入库的坐标，也要分析与这个库有依赖的库的坐标，如果选错了版本，在后面开发过程中会出现不兼容的问题，这会严重阻碍到项目的推进速度。

Spring Boot 框架对Spring框架的缺点进行改善优化，它采取约定优先于配置的核心思想，让开发人员可以全身心的投入到编写逻辑业务代码中，不必在配置和编码之间频繁切换，提高了开发的效率。它不是对Spring框架的增强，而是为我们的开发提供了一个更快的入门方式。Spring Boot的核心功能分别为起步依赖和自动配置。起步依赖可以将具有某种功能的坐标打包，同时提供了一些默认功能；自动配置是一个应用系统启动时的过程，在这个过程中Spring自动考查众多相关因素，并决定配置应该用哪个，不应用哪个。

1. Thymeleaf

Thymeleaf 是在服务器端的一种Java模板引擎，用来Web开发中渲染HTML内容。在Spring框架所支持的所有渲染模版中，不包含jsp，其支持的主要有Velocity、Thymeleaf等。并且Thymeleaf与Spring框架的自动化配置技术可以完美契合，开发人员只需要关注其基本语法，融合它们的成本非常少。

Thymeleaf的特点

动静结合：Thymeleaf最大的特点就是在没有网络的环境下依然可以运行，即不需要启动整个web系统就可以查看浏览器界面的静态效果。Thymeleaf以html为原型，并且可以在html的标签里加入其他的属性生成模版的展示方式。当浏览器解析页面的时候如果遇到没有定义的属性会自动忽略，让html页面静态的运行。在系统的执行过程中，如果有更新的数据返回，Thymeleaf会替换掉对应的静态部分，实现页面的动态展示。

方便使用：Thymeleaf提供了多种方言，可以快速的套用模版实现绑定，编辑等功能。可以实现类似于EL、JSTL等表达式的效果，减少了修改语法、标签等工作。开发过程中开发人员也可以创建自定义方言使用。

容易整合：在创建Spring Boot项目时，只要添加相关依赖。Spring Boot为Thymeleaf提供了相关解析器，并提供一些默认配置。开发人员在使用时和以前操作jsp一样，代码上几乎没有区别，只是标签内模版的语法有区别。

## 4.3 系统功能结构设计

## 4.4 系统数据库设计

设计一个好的数据库表结构，不仅可以提高用户对数据库操作的效率，还可以提高数据库的安全性。关系型数据库是当前比较流行的数据库之一，其采用了关系模型对数据进行组织存储。可以理解为是用二维的表格构成许多数据表，关系型数据库就是由许多数据表以及他们之间的关联组成的一个数据组织。其主要的优点有：

（1） 结构化程度较高

关系型数据库采用数据表的形式，以统一的格式对数据进行存储，数据表之间的连接关系较为简单，结构简单而且易于修改维护。

（2） 设计简单、访问方便

关系型数据库的设计可以不用考虑系统的内部结构，开发人员可以使用一些非过程开发语言进行数据库的设计。提供了索引、更改等操作，可以很方便的访问数据。

（3） 数据之间相互独立

存储的数据与过程因素及物理因素无关，并且在设计数据表时需要至少满足六大范式中的前三范式。这使得数据之间相互独立，依赖较少。

### 4.4.1 数据库设计原则

在设计数据库时需要遵循以下主要原则：

（1） 标准化

设计数据库的时候，有六大范式需要遵循，关系型数据库的设计需要至少满足3NF原则。即数据表的每一列都是不可分割的、一个表中的非主键元素只依赖于主键元素、消除元素之间的传递依赖。表内由主键唯一标识，表之间通过外键连接。

（2） 对数据变化有一定的预期

设计数据库的时候，应该考虑到数据表中每一个字段在系统使用中可能发生的变化，并且在设置其字段大小时应该留足够的空间，避免发生越界错误。

（3） 运用视图技术

设计数据库的时候，一般不通过应用程序直接访问数据库，而是采用视图（本系统采用了SQLyog），这使得数据库设计以及修改更加灵活。

（4） 尽量不使用触发器

设计数据库的时候，使用触发器会使得系统难以调试，应避免使用，尽量寻找代替方式。

### 4.4.2 数据库各表之间关系设计

### 4.4.3 数据库表设计

# 5、访问控制系统的实现

## 5.1 系统开发环境

## 5.2 系统实现